UNITED STATES PATENT AND TRADEMARK OFFICE

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
10/700,729	11/04/2003	Ravisankar V. Pudipeddi	MS306584.1 / MSFTP530US	4944

			EXAMINER
27195	7590	04/06/2007	TO, BAOQUOC N

AMIN. TUROCY & CALVIN, LLP
24TH FLOOR, NATIONAL CITY CENTER
1900 EAST NINTH STREET
CLEVELAND, OH 44114

ART UNIT	PAPER NUMBER
2162	

SHORTENED STATUTORY PERIOD OF RESPONSE	MAIL DATE	DELIVERY MODE
2 MONTHS	04/06/2007	PAPER

Please find below and/or attached an Office communication concerning this application or proceeding.

If NO period for reply is specified above, the maximum statutory period will apply and will expire 6 MONTHS from the mailing date of this communication.

PTOL-90A (Rev. 10/06)

UNITED STATES PATENT AND TRADEMARK OFFICE

MAILED

APR 0 6 2007

Technology Center 2100

BEFORE THE BOARD OF PATENT APPEALS
AND INTERFERENCES

Application Number: 10/700,729
Filing Date: November 04, 2003
Appellant(s): PUDIPEDDI ET AL.

Himanshu S. Amin, Reg. No. 40,894
<u>For Appellant</u>

EXAMINER'S ANSWER

This is in response to the appeal brief filed 12/29/2006 appealing from the Office action

mailed 07/31/2006.

(1) Real Party in Interest

A statement identifying by name the real party in interest is contained in the brief.

(2) Related Appeals and Interferences

The examiner is not aware of any related appeals, interferences, or judicial proceedings which will directly affect or be directly affected by or have a bearing on the Board's decision in the pending appeal.

(3) Status of Claims

Claim 2 was cancelled.

This appeal involves claims 1 and 3-34.

(4) Status of Amendments After Final

No amendment after final has been filed.

(5) Summary of Claimed Subject Matter

The summary of claimed subject matter contained in the brief is correct.

(6) Grounds of Rejection to be Reviewed on Appeal

The appellant's statement of the grounds of rejection to be reviewed on appeal is correct.

(7) Claims Appendix

The copy of the appealed claims contained in the Appendix to the brief is correct.

(8) Evidence Relied Upon

2001/0020245   Golds et al.   9-2001

(9) Grounds of Rejection

The following ground(s) of rejection are applicable to the appealed claims:

Claim Rejections - 35 USC § 101

35 U.S.C. 101 reads as follows:

Whoever invents or discovers any new and useful process, machine, manufacture, or composition of matter, or any new and useful improvement thereof, may obtain a patent therefor, subject to the conditions and requirements of this title.

MPEP 2106 IV. B.2. (b)

A claim that requires one or more acts to be performed defines a process. However, not all processes are statutory under 35 U.S.C. 101. Schrader, 22 F.3d at 296, 30 USPQ2d at 1460. To be statutory, a claimed computer-related process must either: (A) result in a physical transformation outside the computer for which a practical application in the technological arts is either disclosed in the specification or would have been known to a skilled artisan, or (B) be limited to a practical application within the technological arts.

4.   Regarding claims 1 and 3-34 in view of the above cited MPEP section, are not statutory because they merely recite a number of computing steps without producing any tangible result and/or being limited to a practical application within the technological arts.

Claim 1 recited a computer system which does not include any hardware components (e.g. processor and memory) and produce any tangible results.

The depended claims 3-12 and 33 are rejected under the same reason as to claim 1.

Claim 13 recited a computer-implemented method, however, the computing step does not produce any tangible result and/or being limited to a practical application within the technological arts.

The depended claims 14-29 are rejected under the same reason as to claim 13.

As to claim 30, the system including the means to perform the method; however,

the system does not produce the concrete and repeatable results.

The depended claims 30-31 are rejected under the same reason as to claim 30.

Claim Rejections - 35 USC § 102

The following is a quotation of the appropriate paragraphs of 35 U.S.C. 102 that

form the basis for the rejections under this section made in this Office action:

A person shall be entitled to a patent unless –

(e) the invention was described in (1) an application for patent, published under section 122(b), by another filed in the United States before the invention by the applicant for patent or (2) a patent granted on an application for patent by another filed in the United States before the invention by the applicant for patent, except that an international application filed under the treaty defined in section 351(a) shall have the effects for purposes of this subsection of an application filed in the United States only if the international application designated the United States and was published under Article 21(2) of such treaty in the English language.

5.   Claims 1 and 3-34 are rejected under 35 U.S.C. 102(e) as being anticipated by

Golds (Pub. No. US 2001/0020245 A1).

The applied reference has a common assignee with the instant application.

Based upon the earlier effective U.S. filing date of the reference, it constitutes prior art

under 35 U.S.C. 102(e). This rejection under 35 U.S.C. 102(e) might be overcome

either by a showing under 37 CFR 1.132 that any invention disclosed but not claimed in

the reference was derived from the inventor of this application and is thus not the

invention "by another," or by an appropriate showing under 37 CFR 1.131.

Regarding on claims 1 and 34, Golds teaches a computer system that facilitates

management of a file system filter, comprising:

At least one minifilter (filter A) that has an altitude that is an integer value numeric values (.25 and .30) (page 4, paragraph 0035) associated therewith (filter A has an altitude) (page 4, paragraph 0033); and

A filter manager (the filter manage 380) (page 5, lines 1-2, paragraph 0041) that maps altitudes of the at least one minifilter to legacy filter order group (legacy filter drivers are those that do not have a number assigned therefore) (page 5, paragraph 0038).

Regarding on claim 3, Golds teaches the system recited in claim 1, the altitudes are unique values (.25 and .30 are unique value) (page 4, paragraph 0035).

Regarding on claim 4, Golds teaches the system recited in claim 3, the altitudes define the full order of the minifilters with respect to each other (relative order between filters) (page 3, paragraph 0032).

Regarding on claim 5, Golds teaches the system recited in claim 1, multiple instances of the filter manager attach to a file system stack (page 5, lines 0043).

Regarding on claim 6, Golds teaches the system recited in claim 5, each instance of the filter manager attach to a file system stack (page 3, paragraph 0029).

Regarding on claim 7, Golds teaches the system recited in claim 1, the at least one minifilter is code to permit dynamic loading and/or unloading to a filter stack (page 3, paragraph 0030).

Regarding on claim 8, Golds teaches the system recited in claim 7, the altitude of the at least one minifilter ensures that the at least one minifilter, if unloaded, will reload to its previous position in the filter stack (page 5, paragraph 0040).

Regarding on claim 9, Golds teaches the system recited in claim 1, further comprising at least one frame dynamically associated with a single minifilter (page 5, paragraph 0042).

Regarding on claim 10, Golds teaches the system recited in claim 1, further comprising at least one frame dynamically associated with at least one minifilter (page 5, paragraph 0042).

Regarding on claim 11, Golds teaches the system recited in claim 10, further comprising a numerical interval associated with each frame (page 4, lines 0035).

Regarding on claim 12, Golds teaches the system recited in claim 11, the altitude of at least one minifilter has a value within the numerical interval associated with each frame (page 5, paragraph 0040)

Regarding on claims 13 and 35, Golds teaches a computer implemented method for managing a file system filter, comprising:

Loading at least one minifilter to a file system (filter manger request for install the filter driver) (page 5, paragraph 0042); and

Determining an integer altitude value (numeric values .25 and .30) (page 4, paragraph 0035) associated with the at least one minifilter (the manager determines the altitude of the filter driver) (page 5, paragraph 0042).

Regarding on claim 14, Golds teaches a method of claimed 13, further comprising scanning at least one filter manager frame in the file system to find an altitude interval [L, H] associated with the at least one filter manager frame, wherein L is

the lower boundary value of the interval and H is the upper boundary value of the interval (page 5, paragraph 0040).

Regarding on claim 15, Golds teaches the method recited in claim 14, further comprising scanning filter manager frames to determine a frame altitude interval that encompasses the altitude value of the at least one minifilter, such that L<X,H, wherein X is the altitude of the at least one minifilter (page 5, paragraph 0040).

Regarding on claim 16, Golds teaches the method recite in claim 15, further comprising inserting at least one minifilter into the filter manager frame with a corresponding altitude interval upon discovery thereof (page 5, paragraph 0042).

Regarding on claim 17, Golds teaches the method recited in claim 16, further comprising updating minifilter object associated with the at least one minifilter to point to the frame into which the minifilter has been inserted (page 5, paragraph 0042).

Regarding on claim 18, Golds teaches the method recited in claim 15, further comprising scanning filter manager frames for altitude interval, [L1, H1], [L2, H2], adjacent to the altitude value X of the at least one minifilter if no single interval [L, H} encompassing the altitude value X of the at least one minifilter is found, such that the value of the altitude, X, of the at least one minifilter is greater than the upper boundary value of the lower interval H1 and less than the lower boundary value of the higher interval L2 (page 5, paragraph 0042).

Regarding on claim 19, Golds teaches the method recited in claim 18, further comprising:

Inserting the at least one minifilter into the frame having the higher interval (page 5, paragraph 0042);

Adjusting the interval of the frame to [X, H2] (page 5, paragraph 0042); and

Initializing the filter object associated with the at least one minifilter to point to the frame into which the at least one minifilter has been inserted (page 5, paragraph 0042).

Regarding on claim 20, Golds teaches the method recited in claim 18, further comprising creating a new frame and stacking the new frame at the top of the file system stack, if no intervals adjacent to the altitude value of the at least one minifilter are found (page 5, paragraph 0042).

Regarding on claim 21, Golds teaches the method recited in claim 20 further comprising pre-allocating the new frame for management of the at least one minifilter (page 5, paragraph 0042).

Regarding on claim 22, Golds teaches the method recited in claim 21, further comprising calling the filter manager's file system notification routine to submit a request to register for file system notifications (page 5, paragraph 0042).

Regarding on claim 23, Golds teaches the method recited in claim 22, further comprising:

Inserting the minifilter into the new frame (page 5, paragraph 0042);

Initializing the frame interval upper and lower boundary values to the altitude value of the at least one minifilter such that the interval is [H, X] (page 5, paragraph 0042); and

Updating a filter object associated with the at least one minifilter to point to the new frame; wherein the request to register was successful (page 5, paragraph 0042).

Regarding on claim 24, Golds the method recited in claim 22, further comprising:

Removing the new frame from the filter stack (page 5, paragraph 0042)

Extracting the altitude interval from the next lower, now top-most, frame in the stack (page 5, paragraph 0042).

Collapsing the at least one minifilter into the top-most frame (page 5, paragraph 0042); and

Adjusting the frame interval so that the upper boundary value is set equal to the value of the altitude of the at least one minifilter, such that the adjusted interval is [L, X]; wherein the request for registration failed (page 5, paragraph 0042).

Regarding on claim 25, Golds teaches the method recited in claim 22, further comprising determining the identity of a frame calling into the file system notification routine (page 5, paragraph 0042).

Regarding on claim 26, Golds teaches the method recited in claim 25, wherein the identity of the frame is determined by counting the number of all filter manager device objects, N, already in the stack, from top to bottom, using existing application program interface, and wherein each device object represents a frame (page 5, paragraph 0042).

Regarding on claim 27, Golds teaches the method recited in claim 26, further comprising initializing a counter to N and decrementing the counter for every node encountered from the bottom to the top of the stack (page 5, paragraph 0042).

Regarding on clam 28, Golds teaches the method recited in claim 27, wherein a zero value in the counter represents the position of the frame that corresponds to the attachment of the filter manager (page 5, paragraph 0040).

Regarding on claim 29, Golds teaches a computer system that facilitates management of a file system filter, comprising:

Means for mapping integer value altitudes (numeric values .25 and .30) (page 4, paragraph 0035) of minifilters to legacy filter order groups (the assigned altitude within a class will be arbitrary, e.g., there will be combination where A could work above B just well as B could will work above A) (page 4, paragraph 0033); and

Means for determining an altitude interval associated with at least one frame (filter manager determines the altitude of the filter driver) (page. 5, paragraph 42).

Regarding on claim 30, Golds teaches the system recited in claim 29, further comprising means for inserting at least one minifilter into a frame (insert the filter driver) (page 5, paragraph 0042).

Regarding on claim 31, Golds teaches the system recited in claim 30, further comprising means for altering a frame interval to embrace a given minifilter altitude (page 5, paragraph 0042).

Regarding on claim 32, Golds teaches system recited in claim 31, further comprising means for creating a frame for management of at least one minifilter (page 5, paragraph 0042).

(10) Response to Argument

A.   Rejection of claims 1 and 3-34 under 35 U.S.C 101.

The applicant argues that claims 1, 13 and 29 produce one or more useful,

concrete and tangible result. Claim 1 recites: a computer system that facilitates

management of a file system filter, comprising: at least one minifilter that has an integer

altitude value associated therewith; and a filter manager the maps altitude of the at least

one minifilter to legacy filter order groups. As can be seen independent claim 1 provide

a computer system that facilitates management of a file system filter. In order to

facilitate such management of the file system filter, a filter manager maps integer

altitude values associated with at least one minifilter legacy filter order group. It is

submitted that the aforementioned result are useful, concrete and tangible, such results

allow the coexistence between the legacy filters already part of a file system and new

filters, thus mitigating the need for developers of legacy filters to perform substantial

modification on legacy filters in order to ensure coexisting with new filters.

The examiner respectfully disagrees with the above argument. According to the

claim the manager filters only map altitude of the at least one minifilter to a legacy filter

order groups such mapping only draw an association between one minifilter to a legacy

filter and such mapping is not stored in the or make readily available. Therefore, there

is no concrete and tangible result.

Applicant also argues "the subject claim, like independent claim 1, recites useful,

concrete and tangible results, namely, a computer-implemented method for managing a

file system filter, loading at least one minifilter to a file system, and determining an

integer altitude value associated with the at least one loaded minifilter."

The examiner respectfully disagrees with the above argument. The claim

limitation recites "loading at least one minifilter to a file system; and determining an

integer value altitude value associated with the at least one minifilter" which indicate

there is not a integer value altitude being produce associated with the at least one

minifilter. Rather, it is an abstract idea or a thought not yet produces any useful,

concrete and tangible result.

The applicant also argues "independent claim 29 recites one or more useful,

concrete and tangible results, namely: mapping integer value altitude of minifilters to

legacy filter order group; and determining an altitude interval associated with at least

one frame, thereby facilitating management of a file system filter."

The examiner respectfully disagrees with the above argument. As indicate

above, "the mapping is to draw an association between integer value altitude of a

minifilter to legacy filter order group; and determining an altitude interval associated with

at least one frame" is an abstract idea or a thought which does not produce any

concrete, useful and tangible result. There is not an altitude interval being produced nor

the mapping ...and the altitude interval being stored or make readily available.

B.   Rejection of claims 1-34 under 35 U.S.C 102(e).

Applicant argues, "Golds et al. fails to disclose or suggest this aspect of the

invention as claimed in one instance, the claimed subject matter provides a filter

manager that allow coexistence between legacy filters already extant as part of a file

system and newer filters, thereby facilitating management of the priority and/or order of

such filter and their execution. Such a filter manager infrastructure can mitigate the

necessity for developers of legacy filters to perform substantial modification on these

filters because the disclosed management infrastructure seamlessly allows legacy filters

to coexist with newer filters. Accordingly, appellant's claimed subject matter permits

filter managers to insert newer "minifilters" between the filters to crate a new filter

framework. Working within a legacy filter framework, minifilters can be moved as

desired. Moreover, minifilters can dynamically change their own positions as desired,

and such advantageously enhances the ability of a filter manager to sort incoming

requests.

The examiner respectfully disagrees with the above argument. The languages

such as "manager that allow coexistence between legacy filters already extant as part of

a file system and newer filters, thereby facilitating management of the priority and/or

order of such filter and their execution, filter managers to insert newer "minifilters"

between the filters to create a new filter framework and minifilters can be moved as

desired" is not what being claimed "at least one minifilter that has an integer altitude

value associated therewith; and filter manager that maps altitudes of the at least one

minifilter to legacy filter order group."

Applicant argues that "the claimed subject matter in contrast, employ and assign

integer values to facilitate ordering of file system filters, rather than assigning floating

point values to software modules. As would be understood by those of ordinary skill in

the art there is a clear distinction, albeit subtle, between floating point and integer representations. An integer, as would be comprehended by those cognizant in the art, is as a whole number (e.g., a natural number, a number that is neither a fraction nor a mixed fraction). A floating point value in contrast is not an integer as it is representative of numbers with fractional parts to them. It is thus clearly apparent the apparent the apparent that appellant's claimed subject matter and the cited document are distinguishable."

The examiner respectfully disagrees with the above argument. The reference invention uses the integer value to facilitate ordering of the file system which the same concept of the instance application using the float values (paragraph 0030 and paragraph from 0035 to 0036). The float point value, for example rang of number between +2 to –2, the float number including (2.0, 1.9, 1.8, 1.7, 1.6, 1.5, 1.4, 1.3, 1.2, 1.1, 1.0....- 2). The float point values include the integer values. The float point values in the instance application uses to ordering the filters from the existing and the new filters, which serve the same purpose as the prior art. Unless the integer values have a significant to the claimed invention, which has not defined. Therefore, the integer values are used for the same purpose as the float point values.

(11) Related Proceeding(s) Appendix

No decision rendered by a court or the Board is identified by the examiner in the Related Appeals and Interferences section of this examiner's answer.

For the above reasons, it is believed that the rejections should be sustained.
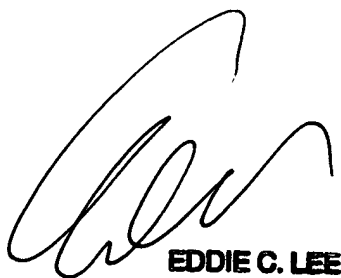
Respectfully submitted,

Examiner

Baoquoc N. To

Conferees:

Eddie Lee

EDDIE C. LEE
SUPERVISORY PATENT EXAMINER

John Breene

JOHN BREENE
SUPERVISORY PATENT EXAMINER
TECHNOLOGY CENTER 2100